

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: James R. Wason

Examiner: Joshua D. Campbell

Serial No: 09/616,809

Art Unit: 2178

Filed: July 14, 2000

Docket: END920000080US1 (13679)

For: TEXT FILE INTERFACE
SUPPORT IN AN OBJECT
ORIENTED APPLICATION

Dated: November 28, 2006

Confirmation No.: 6597

Mail Stop AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REPLY BRIEF

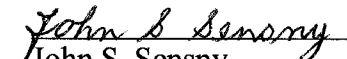
Sir:

Pursuant to 35 U.S.C. 134 and 37 C.F.R. 41.41, entry of this Reply Brief in response to the Examiner's Answer in the above-identified matter, is respectfully requested.

CERTIFICATE OF ELECTRONIC FILING

I hereby certify that this correspondence is being deposited with the United States Patent & Trademark Office via Electronic Filing through the United States Patent and Trademark Office e-business website, on November 28, 2006.

Dated: November 28, 2006


John S. Sensny

I. INTRODUCTION

Appellants have appealed the rejection of Claims 1-3 and 5-17 under 35 U.S.C. 102 as being fully anticipated by U.S. Patent 6,317,871 (Andrews, et al.).

This rejection of these claims should be reversed because Andrews, et al. does not disclose using a macro, which was invoked by one template, both (i) to map data from a text file to a computer application, and (ii) itself to invoke another template to further process the text file. Each of the independent Claims 1, 6 and 10 describe this feature and thus distinguish over Andrews, et al.

II. Discussion

Both the present invention and Andrews, et al. disclose procedures for translating text, such as legacy text, and both Andrews, et al. and the present invention refer to macros. As the Examiner observes from Andrews, et al, a macro is a “complicated function” of several lines of text (Examiner’s Answer, page 5, lines 15 and 16).

There is, however, a critical difference between the present invention and Andrews, et al. Specifically, the present invention uses macros – that is, those “complicated functions” – to map text from a text file to another computer application. Andrews, et al. describes how to translate macros, but does not use the macros to map the translation.

In the Answer, the Examiner refers to an example in Andrews, et al, in which the “symbol ‘ID’ is a macro for a complicated algorithm, thus whenever the symbol ID appears in the program it will be substituted for its definition.” (Examiner’s Answer, page 5, lines 18-20). With this example, though, the macro – that is, the “complicated algorithm” – is not used to map text from a source code to a target code. Instead, the Andrews, et al. procedure involves only substituting a symbol for the macro and its definition.

The Andrews, et al. procedure may use “macro expansion,” but importantly the macro itself is not used to map text from the source code to the target code.

The Examiner also refers to column 4, lines 10-51 of Andrews, et al. for the disclosure that “when the invocation syntax of a macro is recognized, the text which that invocation syntax represents is substituted at that point” (Examiner’s Answer, page 6, lines 12 and 13). In the Andrews, et al. procedure, the macro – that is, the complicated functions or algorithm – is not used to map text, and instead, at a certain point in the text, one item is simply substituted for another item. In fact, Andrews, et al. does not even specifically tell us what the macro is. Of course, with the Andrews, et al. procedure, it does not matter what the macro is, because that procedure is not using the macro, but is only translating the macro.

In the Answer, the Examiner also refers generally several times to columns 7-9 of Andrews, et al. This portion of Andrews, et al. discusses how the partition templates are used and how the Rosetta Translator builds a pTAL fragment tree, translates the contents of a macro body, and pieces together instances of target language partition templates to form target language output files.

Applicant’s Attorney has carefully reviewed Andrews, et al, particularly columns 7-9. As discussed above, in the process described in Andrews, et al, macros are translated, but these macros are not used to map the translation. Andrews, et al. includes several references to translating macros. For example, in Column 8, Lines 41-43, Andrews, et al. notes that “the text of macro actual parameter fragment templates is collapsed into the invoking partition just before the file is pieced together.

In Column 8, Lines 47-50 of Andrews, et al., it is explained that the source generator “fits the text representing the body of the macro stuff into the macro definition.” Further, in Column 8, Lines 57-59 of Andrews, et al., it is explained that “Inconsistently translated code can appear in any partition, not just in macro bodies.” Also, in Column 9, Lines 17-19, Andrews, et al. refer to “exposing different code (which might contain macro definitions) and directives to the translator.” Nowhere in Andrews, et al, though, is there any reference to using the macro to map the translation.

It is critical to emphasize, Appellants respectfully submit, that the Court of Appeals for the Federal Circuit requires a strict identity test in order for a reference to anticipate a claim under 35 U.S.C. 102. For instance, in Apple Computer, Inc. v. Articulate Systems, Inc., 57 USPQ2d 1057, 1061 (Fed. Cir 2000), the Court explained that: “Anticipation under 35 U.S.C. 102 requires the disclosure in a single piece of prior art of each and every limitation of a claimed invention.” “Substantial identity” or “equivalency” is not sufficient. RCA Corp. V. Applied Digital Data Sys., Inc., 221 USPQ 385 (Fed. Cir. 1984).

Independent Claims 1, 6 and 10 of the present application describe how the macros are used in the instant invention. Each of these claims describes the features that a plurality of templates are formed, and that during processing of the text file by one of the templates, a pointer in that template is used to invoke the macro class. These claims also describe the features that this macro class maps data from the text file to the computer application, and then itself is used to invoke another one of the templates to further process the text file. The way in which the first template invokes a macro, which then invokes a second template, is not shown in Andrews, et al.

This feature of the invention is of considerable utility. This nested use of templates and macros allows a processing structure to be built up that mirrors the inherent structure of the text file. Since the behavior of the macro depends both on its internal logic and the template it is passed to invoke, it is possible to reuse the same macro to do different things by passing it a different template. The net effect is that the bulk of the logic needed to describe flow of control may be included in the template structure.

Claims 14-16

In addition to the foregoing, Claims 14, 15 and 16 distinguish over Andrews, et al. because of the additional features described in Claims 14-16. Claim 14 is representative of Claims 15 and 16.

Claim 14 elaborates on the procedure used for a macro to invoke another template. In particular, as described in Claim 14, the name of another template is passed to the macro, and the macro then uses that name to invoke that other template. This results in a nested aggregation of templates and macros.

Andrews, et al, since it does not show using the macro to invoke another template, clearly does not show or suggest any specific procedure, such as the one set forth in Claim 14, for actually invoking that other template. Accordingly, Claim 14 and Claims 15 and 16 independently distinguish over Andrews, et al. because of the features expressly set forth in these claims.

In support of the rejection of Claims 14-16, the Examiner, in the Answer refers generally to column 7, line 65-column 9, line 50, of Andrews, et al, and contends that this portion of this reference discloses passing to the macro the name of the template to be used next (Examiner's Answer, page 4, lines 8-11). A careful study of this portion of Andrews, et al, as well as the rest of the reference, shows that Andrews, et al. does not disclose or teach passing the template name to the macro or using the template name by the macro, as described in Claims 14-16.

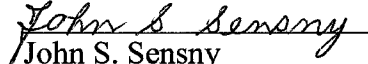
III. Conclusion

Because of the above-discussed differences between Claims 1, 6 and 10 and Andrews, et al, it cannot be said that this reference anticipates any of these claims. Claims 2, 3, 5, 14 and 17 are dependent from, and distinguish over Andrews, et al, with, Claim 1. Claims 7-9 and 15 are dependent from Claim 6 and distinguish therewith over Andrews, et al, and

Claims 11-13 and 16 are dependent from Claim 10 and distinguish therewith over Andrews, et al. In addition, Claims 14-16 separately distinguish over Andrews, et al. because of the additional features expressly described in these claims. Thus, the rejection of the claims over Andrews, et al, under 35 U.S.C. 102, is not proper, and the Board is respectfully requested to reverse this rejection.

Respectfully submitted,

Dated: November 28, 2006


John S. Sensny
Registration No. 28,757
Attorney for Applicants

Scully, Scott, Murphy & Presser, P.C.
400 Garden City Plaza – Suite 300
Garden City, New York 11530
(516) 742-4343

JSS:jy